

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2005-293440

(P2005-293440A)

(43) 公開日 平成17年10月20日(2005.10.20)

(51) Int.Cl.⁷

G06F 5/00

G06F 17/21

F I

G06F 5/00

A

G06F 17/21 570L

テーマコード (参考)

5B009

審査請求 未請求 請求項の数 9 OL (全 15 頁)

(21) 出願番号 特願2004-110704 (P2004-110704)

(22) 出願日 平成16年4月5日(2004.4.5)

(特許庁注：以下のものは登録商標)

1. フロッピー

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(71) 出願人 000232254

日本電気通信システム株式会社

東京都港区三田1丁目4番28号

(74) 代理人 100079164

弁理士 高橋 勇

(72) 発明者 緒方 孝文

東京都港区三田一丁目4番28号 日本電

気通信システム株式会社内

(72) 発明者 新保 和義

東京都港区芝五丁目7番1号 日本電気株

式会社内

Fターム(参考) 5B009 TA11

(54) 【発明の名称】 文字コード変換方法および文字コード変換機能付コンピュータと文字コード変換用プログラム

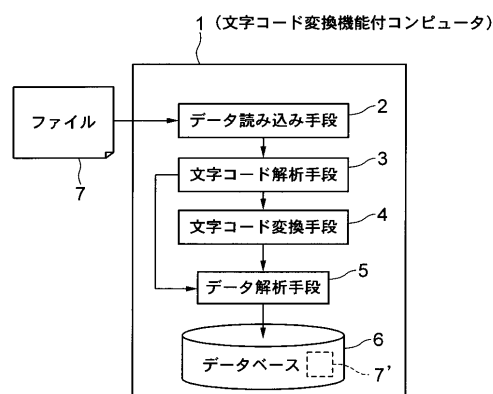
(57) 【要約】

【課題】 オペレーションシステムやコンピュータ識別コード等を指定しなくても適切な文字コード変換を実行することのできる文字コード変換方法を提供する。

【解決手段】 読み込みの対象となるデータの文字コードとして想定される文字コードの特徴を文字コード表1, 2, 3, ..., nとして予めコンピュータ1に記憶させておく。読み込みの対象となるファイル7のデータを記述している文字コードの特徴と文字コード表1, 2, 3, ..., nの特徴を、文字コードに対応する文字の定義の有無に基いてコンピュータ1で比較し、ファイル7を記述している文字コード表jを特定する。文字コード表jとコンピュータ1が使用する文字コード表1との組み合わせに適合したルールを適用してファイル7に公知の文字コード変換を施し、変換後のファイル7'をコンピュータ1上で動作するプログラムに読み込ませる。

。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前に前記データを記述している文字コードをコンピュータのオペレーティングシステムに適合させるための文字コード変換方法であって、

読み込みの対象となるデータの文字コードとして想定される文字コードの特徴を予め前記コンピュータに記憶させておき、

前記データを前記プログラムに読み込ませるための前処理として、前記データを記述している文字コードの特徴と予め前記コンピュータに記憶されている何種類かの文字コードの特徴とを比較して前記データを記述している文字コードを特定し、

10

前記特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データに対して文字コード変換を行い、

文字コード変換によって生成された新たなファイルをプログラム実行用のデータ記述ファイルとして保持することを特徴とした文字コード変換方法。

【請求項 2】

文字コード変換の際に生成された不要な文字列を削除してプログラム実行用のデータ記述ファイルとして保持することを特徴とした請求項 1 記載の文字コード変換方法。

【請求項 3】

不要な文字列として、少なくとも、連続するスペースを削除することを特徴とした請求項 2 記載の文字コード変換方法。

20

【請求項 4】

読み込みの対象となるデータの文字コードとして想定される文字コードの特徴として、文字コード表を記憶しておくことを特徴とした請求項 1、請求項 2 または請求項 3 記載の文字コード変換方法。

【請求項 5】

特徴の比較に際し、データを記述している文字コードの幾つかを抽出し、この文字コードに対応する文字が前記文字コード表に定義されている場合に、前記データを記述している文字コードが当該文字コード表の文字コードであると特定することを特徴とした請求項 4 記載の文字コード変換方法。

30

【請求項 6】

データを記述している文字コードの特徴と予めコンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段と、

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段とを備えたことを特徴とする文字コード変換機能付コンピュータ。

【請求項 7】

データを記述している文字コードの特徴と予めコンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段と、

40

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段と、

前記文字コード変換手段で文字コードを変換されたデータを解析して不要な文字列を削除するデータ解析手段とを備えたことを特徴とする文字コード変換機能付コンピュータ。

【請求項 8】

ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前に前記データを記述している文字コードをコンピュータのオペレーティングシステムに適

50

合させるための文字コード変換用プログラムであって、

前記コンピュータを、前記データを記述している文字コードの特徴と予め前記コンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段、

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段として機能させるための文字コード変換用プログラム。

【請求項 9】

ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前に前記データを記述している文字コードをコンピュータのオペレーティングシステムに適合させるための文字コード変換用プログラムであって、 10

前記コンピュータを、前記データを記述している文字コードの特徴と予め前記コンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段、

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段、

前記文字コード変換手段で文字コードを変換されたデータを解析して不要な文字列を削除するデータ解析手段として機能させるための文字コード変換用プログラム。 20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前にデータを記述している文字コードをコンピュータのオペレーティングシステムに適合させるための文字コード変換方法および文字コード変換機能付コンピュータと文字コード変換用プログラムに関する。

【背景技術】

【0002】

ファイルを使ってプログラムにデータを入力する場合、そこで使用されるファイルは、実際にプログラムが動作するオペレーティングシステムとは別のオペレーティングシステム上の汎用ソフトウェア、例えば、エディタ等で作成されることが多い。 30

また、一般的に、文字コードはオペレーティングシステム毎に異なる。よって、プログラムにデータを入力する際には、実際にプログラムが動作するオペレーティングシステムに合わせ、予め、文字コードを変換したファイルを用意しておく必要があって煩わしい。

【0003】

文字コードの相違によって生じる不都合を改善するための技術としては、例えば、特許文献 1 に開示されるような文字データ変換転送方式、あるいは、特許文献 2 に開示されるような文字コード自動変換装置が公知である。

【0004】

特許文献 1 に開示された文字データ変換転送方式は、オペレーティングシステムの異なるコンピュータ間でのデータの受け渡しに際し、データの転送元で転送先のオペレーティングシステムに合わせてデータの文字コードを変換するものであり、文字コードの変換処理が自動化される点では便利であるが、文字コードの変換処理は、必ず、データの転送元で実行する必要がある。 40

従って、既に何らかの文字コードで記述されたデータがコンピュータの内部または外部に存在したとしても、データを記述している文字コードが当該コンピュータのオペレーティングシステムの文字コードと相違すれば、このデータを当該コンピュータ上で動作するプログラムに読み込ませることはできない。

つまり、コンピュータ上で動作するプログラムにデータを入力する際には、実際にプロ 50

グラムが動作するコンピュータとは異なる別のコンピュータを利用して予め文字コード変換済みのファイルを準備しておかなければならず、最低限、コンピュータが2台は必要となる欠点がある。

【0005】

特許文献2に開示された文字コード自動変換装置も、取り扱う文字コードの異なる2以上のコンピュータ間でデータの整合性を得るための文字コード変換を行うといった技術思想に関しては特許文献1のものと同様であり、従って、前記と同様の不都合がある。

【0006】

また、何れのものも、オペレーションシステムの名称やコンピュータ識別コード等を利用者自身が指定することで変換元や変換先の文字コードを特定する構成であるため、コンピュータの操作ミスにより適切な文字コード変換が行えなくなる欠点がある。 10

【0007】

【特許文献1】特開平4-361323号公報

【特許文献2】特開平5-341956号公報

【発明の開示】

【発明が解決しようとする課題】

【0008】

そこで、本発明の課題は、前記従来技術の不都合を改善し、既に何らかの文字コードで記述されたデータがコンピュータの内部または外部に存在した場合に、単一のコンピュータにより、このデータを記述している文字コードをコンピュータのオペレーティングシステムに適合させることができ、また、オペレーションシステムの名称やコンピュータ識別コード等を利用者がわざわざ指定しなくても適切な文字コード変換を実行することのできる文字コード変換方法および文字コード変換機能付コンピュータと文字コード変換用プログラムを提供することにある。 20

【課題を解決するための手段】

【0009】

本発明の文字コード変換方法は、ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前に前記データを記述している文字コードをコンピュータのオペレーティングシステムに適合させるための文字コード変換方法であり、前記課題を解決するため、特に、 30

読み込みの対象となるデータの文字コードとして想定される文字コードの特徴を予め前記コンピュータに記憶させておき、

前記データを前記プログラムに読み込ませるための前処理として、前記データを記述している文字コードの特徴と予め前記コンピュータに記憶されている何種類かの文字コードの特徴とを比較して前記データを記述している文字コードを特定し、

前記特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データに対して文字コード変換を行い、

文字コード変換によって生成された新たなファイルをプログラム実行用のデータ記述ファイルとして保持することを特徴とした構成を有する。 40

【0010】

以上の構成により、ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませるための前処理として、まず、データを記述している文字コードの特徴と予めコンピュータに記憶されている何種類かの文字コードの特徴とを比較し、データを記述している文字コードを特定する。

この際、オペレーションシステムの名称やコンピュータ識別コード等を指定してデータを記述している文字コードやコンピュータのオペレーティングシステムで使用されている文字コードを明示する必要はなく、データを記述している文字コードの特徴と予めコンピュータに記憶されている何種類かの文字コードの特徴とを直接的に比較することでデータを記述している文字コードが特定されるので、利用者側におけるコンピュータの操作ミス等 50

によって文字コードの指定を誤るといった問題が解消される。

このようにして、データを記述している文字コードが特定された後、特定された文字コードで記述されているデータがコンピュータのオペレーティングシステムで使用されている文字コードで記述されるようにデータに対する文字コード変換が行われて新たなファイルが生成され、このファイルがプログラム実行用のデータ記述ファイルとして保持される。

従って、データ記述ファイルをコンピュータ上で動作するプログラムに読み込ませることで、当該プログラムに固有の処理操作を的確に実行させることができる。

また、データに対する文字コード変換はプログラムが動作するコンピュータ上で実行されるので、文字コード変換のために別途コンピュータを準備する必要はなく、何らかの文字コードで記述されたデータが既に存在すれば、このデータを其のまま使用してプログラムを実行させることができる。

【 0 0 1 1 】

更に、文字コード変換の際に生成された不要な文字列を削除してプログラム実行用のデータ記述ファイルとして保持するようにしてもよい。

【 0 0 1 2 】

プログラムの実行前に予め不用な文字列を削除しておくことにより、プログラムの誤動作やプログラムの実行に伴う異常の発生を未然に防止することができる。

【 0 0 1 3 】

この際、不要な文字列として、少なくとも、連続するスペースを削除することが望ましい。

【 0 0 1 4 】

データを記述している文字コードが適切に特定された場合であっても、例えば、キャリッジリターン (C R) やラインフィード (L F) 等のように適切な置き換えが難しい文字制御コードがあり、意味のないスペースが連続的に生成される場合がある。連続するスペースを削除することで、置き換えが困難な文字制御コードに起因する異常を軽減することができる。

【 0 0 1 5 】

また、読み込みの対象となるデータの文字コードとして想定される文字コードの特徴としては、文字コード表を記憶させておくことが望ましい。

【 0 0 1 6 】

文字コードの相違は本質的には文字コード表の相違であるから、データを記述している文字コードの特徴と何種類かの文字コード表の特徴とを比較すれば、データを記述している文字コードを特定することが可能である。

【 0 0 1 7 】

具体的には、データを記述している文字コードの幾つかを抽出し、この文字コードに対応する文字が文字コード表に定義されている場合に、データを記述している文字コードが当該文字コード表の文字コードであると特定することができる。

【 0 0 1 8 】

例えば、データを記述している文字コードを先頭から幾つか抽出し、その文字コードに対応する文字が全て文字コード表に定義されている場合、つまり、データを記述している文字コードが文字コード表の未定義箇所を指定していない場合に、データ記述用の文字コードが当該文字コード表の文字コードであると特定することが可能である。

【 0 0 1 9 】

また、本発明の文字コード変換機能付コンピュータは、前記と同様の課題を達成するため、

データを記述している文字コードの特徴と予めコンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段と、

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コ

ンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段とを備えたことを特徴とする構成を有する。

【0020】

以上の構成において、文字コード解析手段が、データを記述している文字コードの特徴と予めコンピュータに記憶されている文字コードの特徴とを比較し、データを記述している文字コードを特定する。

具体的には、文字コード解析手段が、データを記述している文字コードの幾つかを抽出し、その文字コードに対応する文字が何れの文字コード表で定義されているかを判定し、抽出された文字コードの全てが定義された文字コード表に対応する文字コードをデータの記述に用いられた文字コードとして特定することが可能である。

10

文字コード変換手段は、文字コード解析手段で特定された文字コードで記述されているデータがコンピュータのオペレーティングシステムで使用されている文字コードで記述されるようにデータの文字コードを変換する。

【0021】

また、前記構成に加え、文字コード変換手段により文字コードを変換されたデータを解析して不要な文字列を削除するデータ解析手段を備えるようにしてもよい。

【0022】

プログラムの実行前に予め不用な文字列を削除しておくことにより、プログラムの誤動作やプログラムの実行に伴う異常の発生を未然に防止することができる。

20

データ解析手段は、不要な文字列として、少なくとも、連続するスペースを削除することが望ましい。

【0023】

本発明の文字コード変換用プログラムは、ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませる前に前記データを記述している文字コードをコンピュータのオペレーティングシステムに適合させるための文字コード変換用プログラムであり、前記と同様の課題を達成するため、

前記コンピュータを、前記データを記述している文字コードの特徴と予め前記コンピュータに記憶されている文字コードの特徴とを比較して前記データを記述している文字コードを特定する文字コード解析手段、

30

前記文字コード解析手段で特定された文字コードで記述されている前記データが前記コンピュータのオペレーティングシステムで使用されている文字コードで記述されるように前記データの文字コードを変換する文字コード変換手段として機能させることを特徴とした構成を有する。

【0024】

この文字コード変換用プログラムを実装したコンピュータの作用は既に述べた通りである。

【0025】

また、この文字コード変換用プログラムにより、前記コンピュータを、更に、文字コード変換手段で文字コードを変換されたデータを解析して不要な文字列を削除するデータ解析手段として機能させるようにしてもよい。

40

【0026】

この文字コード変換用プログラムを実装したコンピュータの作用は既に述べた通りである。

【発明の効果】

【0027】

本発明は、ファイルに記述されたデータをコンピュータ上で動作するプログラムに読み込ませるための前処理として、データを記述している文字コードの特徴と予めコンピュータに記憶されている何種類かの文字コードの特徴とを比較してデータを記述している文字コードを特定するようにしたので、オペレーションシステムの名称やコンピュータ識別コー

50

ド等を指定してデータを記述している文字コードやコンピュータのオペレーティングシステムで使用されている文字コードを明示する必要がない。従って、利用者側におけるコンピュータの操作ミス等によって文字コードの指定を誤るといった問題が解消され、適切な文字コード変換を行うことができる。

しかも、文字コード変換はプログラムが動作するコンピュータ上で実行されるので、文字コード変換のために別途コンピュータを準備する必要はなく、何らかの文字コードで記述されたデータが既に存在すれば、このデータを其のまま使用して当該コンピュータ上でプログラムを実行させることができる。

【発明を実施するための最良の形態】

【0028】

10

次に、本発明を実施するための最良の形態について、一例を挙げて説明する。図1は、本発明の文字コード変換方法を適用した文字コード変換機能付コンピュータ（以下、単にコンピュータという）の構成の概略を示した機能ブロック図である。

【0029】

このコンピュータ1は、図1に示される通り、ファイル7に記述されたデータを読み込むためのデータ読み込み手段2と、ファイル7のデータを記述している文字コードの特徴と予めコンピュータ1に記憶されている何種類かの文字コードの特徴とを比較してファイル7のデータを記述している文字コードを特定する文字コード解析手段3を備える。

【0030】

また、文字コード解析手段3で特定された文字コードで記述されているデータがコンピュータ1のオペレーティングシステムで使用されている文字コードで記述されるようにデータの文字コードを変換する文字コード変換手段4と、文字コード変換手段4で文字コードを変換されたデータを解析して不要な文字列を削除するデータ解析手段5を備える。

20

【0031】

文字コード変換手段4による文字コード変換を受けた上でデータ解析手段5によって不要な文字列を削除されたデータは、プログラム実行用のデータ記述ファイルとしてデータベース6内に保持されるようになっている。

【0032】

図2はコンピュータ1の実質的な構成について示したブロック図である。コンピュータ1は、図2に示される通り、演算処理用のCPU8、該CPU8の駆動制御プログラムを格納したROM9、演算データの一時記憶等に用いられるRAM10、大容量の不揮発性記憶手段として機能するハードディスクドライブ11、他のコンピュータや各種ネットワーク等に接続するためのインターフェイス12を有する。そして、CPU8の入出力回路13には、マン・マシン・インターフェイスを構成するディスプレイ14とキーボード15およびマウス16が接続され、更に、データ記憶媒体を介して外部のファイル7を読み込むためのフロッピーディスクドライブ17等が接続されている。

30

【0033】

ハードディスクドライブ11には、本発明を適用した文字コード変換用プログラムが格納されており、この文字コード変換用プログラムをRAM10に読み出してCPU8に実行させることにより、CPU8が、データ読み込み手段2、文字コード解析手段3、文字コード変換手段4あるいはデータ解析手段5として機能するようになっている。

40

【0034】

ハードディスクドライブ11には、コンピュータ1上で動作する他のアプリケーションプログラムとして、例えば、ワープロソフト等が格納されている。

【0035】

また、このハードディスクドライブ11は、プログラム実行用のデータ記述ファイルを保持するためのデータベース6として機能する他、読み込みの対象となるデータの文字コードとして想定される文字コードの特徴を記憶した記憶手段としても機能する。

【0036】

図3、図4、図5は、ハードディスクドライブ11に予め記憶させておいた文字コード

50

表の一例を示した概念図である。

【0037】

ここでは、便宜上、図3の文字コード表を文字コード表1、図4の文字コード表を文字コード表2、図5の文字コード表を文字コード表3と呼ぶことにする。

【0038】

このうち、コンピュータ1のオペレーティングシステムが実際に使用している文字コードは図3の文字コード表1に対応した文字コードであり、図4の文字コード表2と図5の文字コード表3は、読み込みの対象となるデータの文字コードとして想定される文字コードの特徴として記憶された文字コード表である。

【0039】

図3，図4，図5は、何れも2バイト文字コード表の例であり、上位1バイト(H)の数値と下位1バイト(L)の数値からなる文字コード(H，L)で文字コード表のスポット(H，L)を指定することで、各文字コード表に固有の文字を指定するように構成されている。

【0040】

ここで、仮に、図5の文字コード表3の文字コード(H，L)=(0，3)が文字「0」を示すものであると仮定すると、図5の文字コード表3を使用している別のコンピュータ上で動作するプログラム、例えば、ワープロソフトで作成されたデータで文字「0」を表示するために文字コード(H，L)=(0，3)が使用されていた場合、この文字コード(H，L)=(0，3)をコンピュータ1上で動作するワープロソフトに其のまま読み込むと、このコンピュータ1のオペレーティングシステムが使用する図3の文字コード表1で文字コード(H，L)=(0，3)が指定されることになるが、図3の文字コード表1では文字コード(H，L)=(0，3)の文字は未定義であるから、最初に意図した文字「0」が表示されないか、あるいは、エラーが検出されてコンピュータ1上で動作するワープロソフトの動作が停止するといった問題が生じる場合がある。つまり、文字コードの相違から文字コード自体の解釈が不能となる場合がある。

【0041】

また、仮に、図4の文字コード表2の文字コード(H，L)=(3，2)が文字「#」を示し、図3の文字コード表1の文字コード(H，L)=(3，2)が文字「2」を示すものであると仮定すると、図4の文字コード表2を使用している別のコンピュータ上で動作するプログラムで作成されたデータで文字「#」を表示するために文字コード(H，L)=(3，2)が使用されていた場合、この文字コード(H，L)=(3，2)をコンピュータ1上で動作するワープロソフトに其のまま読み込むと、このコンピュータ1のオペレーティングシステムが使用する図3の文字コード表1で文字コード(H，L)=(3，2)が指定され、結果的に、最初に意図した文字「#」ではなく文字「2」が表示されるといった問題が生じる。つまり、文字コードの解釈は可能であっても、文字コードの相違から意図した文字が表示されなくなる場合がある。

【0042】

この場合、実際には、図3の文字コード表1においては文字「#」が文字コード(H，L)=(2，3)で定義されているので、図4の文字コード表2を使用している別のコンピュータ上で動作するワープロソフトで作成されたデータをコンピュータ1上で動作するワープロソフトに読み込む際に、文字コード(H，L)=(3，2)の指定を文字コード(H，L)=(2，3)の指定に変換する処理を行えば、別のコンピュータ上で動作するワープロソフトで作成された文字「#」をコンピュータ1上で動作するワープロソフトにおいても文字「#」として意図した通りに適切に表示することが可能である。

【0043】

このようにして、異なる文字コード表を使用するコンピュータ間あるいはアプリケーション間で、文字コード表の組み合わせに適合した特定のルールを適用して文字コード変換を行う技術については既に公知であるが、この種の技術を適用して適切な文字コード変換を行うためには、各々のコンピュータあるいはアプリケーションが使用している文字コー

10

20

30

40

50

ドが明らかになっていなければならないといった制限がある。また、文字コードが明らかになっていたとしても、各々のコンピュータあるいはアプリケーションが使用している文字コード、あるいは、文字コードを特定するために必要とされるオペレーションシステムの名称やコンピュータ識別コード等をコンピュータに教示するのは利用者自身の役目となるので、利用者がコンピュータの操作をミスすれば、適切な文字コード変換は行えなくなってしまう。

【0044】

こういった不都合を解消するため、この実施の形態では、データを記述している文字コードの特徴と予めコンピュータ1に記憶されている何種類かの文字コードの特徴とをコンピュータ1によって比較させ、データを記述している文字コードをコンピュータ1の内部処理によって特定させるようにしている。 10

【0045】

次に、コンピュータ1上で動作するワープロソフト等のアプリケーションプログラムに他のコンピュータで作成されたファイル7のデータを読み込ませる際にCPU8が実行する前処理の概略を示した図7、図8のフローチャートを参照して、データ読み込み手段2、文字コード解析手段3、文字コード変換手段4、データ解析手段5として機能するCPU8の処理動作について具体的に説明する。

【0046】

前述した通り、ハードディスクドライブ11には、本発明を適用した文字コード変換用プログラムが格納されており、この文字コード変換用プログラムをRAM10に読み出してCPU8が実行することで、この前処理が開始され、CPU8が、データ読み込み手段2、文字コード解析手段3、文字コード変換手段4あるいはデータ解析手段5として機能することになる。 20

【0047】

前処理が開始されると、まず、データ読み込み手段2として機能するCPU8が、指定されたファイル7から文字コードで記述された一連のデータを読み込んでRAM10にバックアップとして一時記憶する(ステップS1)。

【0048】

元データであるファイル7自体はコンピュータ1の内部(ハードディスクドライブ11内)に存在しても外部に存在しても構わないが、外部からの読み込みが必要な場合には、インターフェイス12を使用した他のコンピュータとのネットワーク接続、あるいは、ファイル7を収めたデータ記憶媒体とフロッピーディスクドライブ17等を利用するものとする。 30

【0049】

次いで、CPU8は、文字コード表選択指標jを0に初期化し(ステップS2)、該指標jの値を改めて1インクリメントした後(ステップS3)、ヒット数カウンタkの値を0に初期化する(ステップS4)。

【0050】

そして、文字コード解析手段3として機能するCPU8が、RAM10に一時記憶されたファイル7の先頭からデータを記述している文字コードを1つ読み込み(ステップS5)、文字コード表選択指標jの現在値で特定される文字コード表jを検索して、当該1文字コードが文字コード表jで定義されているか否かを判定する(ステップS6)。 40

【0051】

仮に、現時点でj=1、また、ステップS5の処理で読み込んだ1文字コードが(H, L)=(3, 2)であったとすれば、図3の文字コード表1(=j)の(H, L)=(3, 2)には文字「2」が定義されているので、ステップS6の判定結果は真となる。この場合、ファイル7のデータを記述している文字コード表が図3、図4、図5の何れのものであってもステップS6の判定結果は真となるので、ファイル7のデータの記述に用いられた文字コードが不明な現段階では、データを記述している文字コード(H, L)=(3, 2)によって文字「2」、文字「#」の何れが指定されるのかは特定し得ない。要する 50

に、この時点で問題となるのは、ステップ S 5 の処理で読み込んだ 1 文字コード (H, L) によって指定される文字コード表 j の (H, L) スポットに何らかの文字が定義されているか否かのみである。

【0052】

ここで、ステップ S 6 の判定結果が真となって当該 1 文字コードが文字コード表 j で定義されていることが明らかとなった場合には、文字コード解析手段 3 として機能する CPU 8 がヒット数カウンタ k の値を 1 インクリメントし (ステップ S 7)、該カウンタ k の現在値がコード識別基準値 m に達しているか否かを判定する (ステップ S 8)。

【0053】

コード識別基準値 m は、ファイル 7 のデータを記述している文字コードが、文字コード表選択指標 j で特定される文字コード表 j と一致するか否かを判定するための基準値である。

【0054】

ステップ S 8 の判定結果が偽となってヒット数カウンタ k の現在値がコード識別基準値 m に達していないことが明らかとなった場合には、文字コード解析手段 3 として機能する CPU 8 は、RAM 10 に一時記憶されているファイル 7 から改めて次の文字コードを 1 つ読み込む (ステップ S 5)。そして、当該 1 文字コードが文字コード表 j で定義されていないことがステップ S 6 の判定処理で確認されるか、あるいは、カウンタ k の現在値がコード識別基準値 m に達したことがステップ S 8 の処理で確認されるまでの間、ヒット数カウンタ k の値を逐次インクリメントしながら、前記と同様に、ステップ S 5 ~ ステップ S 8 の処理を繰り返し実行する。

【0055】

このような処理を繰り返し実行する間にステップ S 6 の判定結果が偽となった場合には、ファイル 7 のデータが文字コード表 j で定義されていない文字コードを使用して記述されていること、つまり、ファイル 7 のデータが文字コード表 j 以外の文字コード表を使用して記述されていることを意味する。

【0056】

従って、文字コード解析手段 3 として機能する CPU 8 は、文字コード表選択指標 j の現在値が、ハードディスクドライブ 11 に予め記憶させておいた文字コード表の総数 n (= 3) に達しているか否か、即ち、ファイル 7 のデータを記述している文字コードの特徴と予めコンピュータ 1 に記憶させておいた文字コード表の特徴との比較が全て完了しているか否かを判定することになる (ステップ S 9)。

【0057】

ステップ S 9 の判定結果が真となった場合には、ファイル 7 のデータを記述している文字コードの特徴と予めコンピュータ 1 に記憶させておいた幾つかの文字コード表の特徴との比較が全て完了しており、しかも、ファイル 7 のデータを記述している文字コードの特徴が予めコンピュータ 1 に記憶させておいた何れの文字コード表の特徴とも一致しないことを意味する。この場合、読み込みの対象となるデータの文字コードとして想定されていなかった文字コードによってファイル 7 のデータが記述されていることになるので、文字コードの変換は不能となり、文字コード変換に関わる前処理はこの時点で中断される。

【0058】

一方、ステップ S 9 の判定結果が偽となった場合には、ファイル 7 のデータを記述している文字コードの特徴との比較が行われていない文字コード表が未だ残っていることを意味し、ファイル 7 のデータを記述している文字コードの特徴が他の文字コード表の特徴と一致する可能性があるので、文字コード解析手段 3 として機能する CPU 8 は、文字コード表選択指標 j の値を 1 インクリメントし、該指標 j の現在値に基づいて別の文字コード表 j を改めて選択し直し (ステップ S 3)、ヒット数カウンタ k の値を 0 に初期化した後 (ステップ S 4)、前記と同様にステップ S 5 ~ ステップ S 8 の処理を繰り返し実行する。

【0059】

10

20

30

40

50

そして、このような処理を繰り返し実行する間にステップ S 8 の判定結果が真となった場合には、R A M 1 0 にバックアップとして記憶されたファイル 7 の先頭から順に読み込まれた文字コードが、少なくとも先頭から連続して m 個だけ文字コード表 j に定義されていることを意味するので、文字コード解析手段 3 として機能する C P U 8 は、ファイル 7 のデータが文字コード表 j の文字コードで記述されているものと判定し、前述のループ処理を終了する。

【 0 0 6 0 】

図 3 , 図 4 , 図 5 に例示したように、文字コード表は、その種別によって文字が未定義となっているスポット (H , L) の分布状態が相違するので、ファイル 7 のデータを記述している文字コードの幾つか、例えば、先頭から m 個を抽出し、抽出された文字コードの 10
全てが定義された文字コード表を求めれば、ファイル 7 のデータが何れの文字コード表に基いて生成されたものであるのかを知ることができる。なお、十分な確度で文字コード表を特定するに足るコード識別基準値 m の値は予め求めておくものとする。

【 0 0 6 1 】

以上のようにしてファイル 7 のデータを記述している文字コードに対応した文字コード表 j を特定した後、C P U 8 は、文字コード表選択指標 j の現在値が 1 であるか否か、即ち、ファイル 7 のデータを記述している文字コードとコンピュータ 1 のオペレーティングシステムが実際に使用している文字コードとが同じものであるか否かを判定する (ステップ S 1 0) 。

【 0 0 6 2 】

ステップ S 1 0 の判定結果が真となった場合には、ファイル 7 のデータを記述している文字コードとコンピュータ 1 のオペレーティングシステムが実際に使用している文字コードとが同じものであり、共に文字コード表 1 に基くものであることを意味するので、実質的な文字コード変換は不要であり、このファイル 7 が、R A M 1 0 内に其のまま保持されることになる。 20

【 0 0 6 3 】

一方、ステップ S 1 0 の判定結果が偽となった場合には、ファイル 7 のデータを記述している文字コードとコンピュータ 1 のオペレーティングシステムが実際に使用している文字コードとが相違することを意味するので、文字コード変換手段 4 として機能する C P U 8 は、ファイル 7 のデータを記述している文字コード表 j とコンピュータ 1 のオペレーティングシステムが実際に使用している文字コード表 1 との組み合わせに適合したルールを適用して公知の文字コード変換を実行することで、文字コード表 j に対応した文字コードで記述されているファイル 7 のデータが、コンピュータ 1 のオペレーティングシステムで使用する文字コード表 1 の文字コードで記述されるようにファイル 7 におけるデータの文字コードを変換し (ステップ S 1 1) 、文字コード変換されたファイル 7 を改めて R A M 1 0 に上書きして一時記憶する (ステップ S 1 2) 。

【 0 0 6 4 】

この際、ステップ S 1 の処理でバックアップとして R A M 1 0 に読み込まれた作業用のファイル 7 のデータは失われるが、コンピュータ 1 の内部 (ハードディスクドライブ 1 1 内) あるいはファイル 7 を収めたデータ記憶媒体内の元データはそのまま保持される。 40

【 0 0 6 5 】

次いで、データ解析手段 5 として機能する C P U 8 が、R A M 1 0 に一時記憶されている文字コード変換後のファイル 7 の先頭からデータを記述している文字コードを 1 つ読み込み (ステップ S 1 3) 、当該文字コードがスペースを表すものであるか否かを判定する (ステップ S 1 4) 。

【 0 0 6 6 】

文字コードがスペースを表すものでなければ、C P U 8 は、連続スペース記憶フラグ F をリセットし (ステップ S 1 5) 、この文字コードをファイル 7 に対応するプログラム実行用のデータ記述ファイル 7 ' の適正なデータとしてハードディスクドライブ 1 1 のデータベース 6 内に保存する (ステップ S 1 8) 。

【 0 0 6 7 】

一方、ステップ S 1 4 の判定結果が真となって当該文字コードがスペースを表すものであることが確認された場合には、CPU 8 は、更に、この時点で既に連続スペース記憶フラグ F がセットされているか否か、つまり、今回読み込まれたスペースが連続する 2 個目以降のスペースであるか否かを判定することになる（ステップ S 1 6 ）。

【 0 0 6 8 】

ステップ S 1 6 の判定結果が偽となった場合には、今回読み込まれたスペースが最初のスペースもしくは通常の文字に連続する 1 個目のスペースであることを意味するので、CPU 8 は、連続スペース記憶フラグ F をセットした後（ステップ S 1 7 ）、このスペースが有効な文字列であると見做して、前記と同様に、このスペースを表している文字コードをファイル 7 に対応するプログラム実行用のデータ記述ファイル 7 ' の適正なデータとして、ハードディスクドライブ 1 1 のデータベース 6 内に続けて保存する（ステップ S 1 8 ）。

10

【 0 0 6 9 】

これに対し、ステップ S 1 6 の判定結果が真となった場合には、今回読み込まれたスペースが連続する 2 個目以降のスペースであることを意味するので、データ解析手段 5 として機能する CPU 8 は、このスペースが不要な文字列であると見做してステップ S 1 8 の処理をスキップし、ファイル 7 に対応するプログラム実行用のデータ記述ファイル 7 ' への当該文字コードの保存を禁止する。

【 0 0 7 0 】

次いで、CPU 8 は、今回読み込まれた文字コードがファイル 7 に記述されている最後の文字コードであるか否かを判定し（ステップ S 1 9 ）、最後の文字コードでなければ、RAM 1 0 に一時記憶されている文字コード変換後のファイル 7 から次のデータを記述している文字コードを 1 つ読み込む（ステップ S 1 3 ）。そして、ファイル 7 に記述されている最後の文字コードがステップ S 1 9 の処理で検出されるまでの間、データ解析手段 5 として機能する CPU 8 が、前記と同様にして、ステップ S 1 3 ~ ステップ 1 8 の処理を繰り返し実行する。

20

【 0 0 7 1 】

従って、文字コード変換されたファイル 7 に 2 以上のスペースが連続して存在する場合には、最初のスペースを記述している文字コードのみが有効な文字列として認識され、これに連続したスペースを記述する文字コードはデータ解析手段 5 により不要な文字列としてファイル 7 から削除され、不要な文字列の削除されたファイル 7 が、プログラム実行用のデータ記述ファイル 7 ' としてデータベース 6 内に保存されることになる。

30

【 0 0 7 2 】

一般に、コンピュータのオペレーティングシステムやワープロソフト等のアプリケーションプログラムの相違あるいは文字コードの相違等により、改行処理にラインフィード（LF）を利用するもの、キャリッジリターン（CR）を利用するもの、更には、ラインフィード（LF）とキャリッジリターン（CR）を組み合わせるものがあり、これらの間で文字コード変換を行うと、実質的な意味のない多数のスペースから成る文字列が生成される場合があるが、このようにして連続するスペースを削除することによって不要な文字列を削除することができる。

40

【 0 0 7 3 】

最終的に、適切な文字コード変換を施し且つ不要な文字列を削除したプログラム実行用のデータ記述ファイル 7 ' を元データであるファイル 7 に代えて当該コンピュータ 1 上で動作するアプリケーションプログラム、例えば、ワープロソフト等に読み込ませることにより、ファイル 7 のデータがどのような文字コードで記述されていたかに関わりなく、当該コンピュータ 1 上で作成したファイルの場合と同様に、ワープロソフト等のアプリケーションプログラムを利用した処理操作を的確に実行させることができる。

【 0 0 7 4 】

この実施の形態では、ファイル 7 に対する文字コード変換が、ワープロソフト等のアプ

50

リケーションプログラムを動作させるコンピュータ 1 上で実行されるので、文字コード変換のために別途コンピュータを準備する必要はなく、何らかの文字コードで記述されたファイル 7 が既に存在すれば、前述の前処理をファイル 7 に施すだけで、このファイルを其のまま使用してプログラムを確実に実行させることができる。

【 0 0 7 5 】

また、アプリケーションプログラムの実行前に予め不用な文字列を削除しておくようにしているので、アプリケーションプログラムの誤動作やアプリケーションプログラムの実行に伴う異常の発生も未然に防止することができる。

【 0 0 7 6 】

図 7 に示されるようなデータ解析手段 5 の処理は、当該コンピュータ 1 で文字コード変換を行った場合（ステップ S 1 0 の判定結果が偽となった場合）、あるいは、他のコンピュータで文字コード変換を施して文字コードを整合したファイルを当該コンピュータ 1 に持ち込んだ場合（ステップ S 1 0 の判定結果が真となった場合）の何れにおいても実行されるので、他のコンピュータで文字コード変換を施したファイル 7 から不要な文字列を削除するといったデバッグ処理にも利用可能である。

【 0 0 7 7 】

ここでは、一例として、スペースの連続を不用意な文字列と認識して削除する場合について述べたが、コンピュータ 1 上で動作するアプリケーションプログラムの特性、つまり、プログラム実行用のデータ記述ファイル 7 ' を読み込むプログラムの特性に応じ、他の文字列を不要な文字列として削除したり、更には、データフォーマットの変換等を行うように設計してもよい。

【図面の簡単な説明】

【 0 0 7 8 】

【図 1】本発明の文字コード変換方法を適用した文字コード変換機能付コンピュータの構成の概略を示した機能ブロック図である。

【図 2】文字コード変換機能付コンピュータの実質的な構成について示したブロック図である。

【図 3】文字コード変換機能付コンピュータのオペレーティングシステムが使用している文字コード表の一例を示した概念図である。

【図 4】読み込みの対象となるデータの文字コードとして想定される文字コードの特徴として記憶された文字コード表の一例を示した概念図である。

【図 5】読み込みの対象となるデータの文字コードとして想定される文字コードの特徴として記憶された文字コード表の他の一例を示した概念図である。

【図 6】文字コード変換機能付コンピュータの CPU が実行する前処理（文字コード変換用プログラム）の概略を示したフローチャートである。

【図 7】文字コード変換機能付コンピュータの CPU が実行する前処理（文字コード変換用プログラム）の概略を示したフローチャートの続きである。

【符号の説明】

【 0 0 7 9 】

1 文字コード変換機能付コンピュータ

2 データ読み込み手段

3 文字コード解析手段

4 文字コード変換手段

5 データ解析手段

6 データベース

7 ファイル

8 CPU（文字コード解析手段，文字コード変換手段，データ解析手段）

9 ROM

10 RAM

11 ハードディスクドライブ

40

50

